Paper No. 18

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte BRIAN J. RODDY

Appeal No. 1999-1452
Application 08/436,830

ON BRIEF

Before THOMAS, KRASS and FLEMING, Administrative Patent Judges.

THOMAS, Administrative Patent Judge.

DECISION ON APPEAL

Appellant has appealed to the Board from the examiner's final rejection of claims 1-27, which constitute all the claims in the application.

Representative claim 1 is reproduced below:

1. An object-oriented computing system, comprising:

a data file structure stored in a memory which contains an identification of objects in a software program, properties and handlers associated with respective objects, pointers to ancestor objects from which a given object inherits properties and handlers, and pointers to descendant objects which inherit properties and handlers from a given object;

a user interface via a user selects an object and which displays ancestor objects and descendant objects for the selected object;

means associated with said interface for enabling a user to indicate a change in relationship between a selected object and at least one ancestor or descendant object; and

means responsive to an indicated change for modifying one or more pointers associated with the selected object in said data file structure to correspond to said change.

The following references are relied on by the examiner:

Pazel	5,410,648	Apr. 25, 1995
Frid-Nielsen	5,432,903	July 11, 1995 (filing date Apr. 26, 1994)
Berry	5,546,519	Aug. 13, 1996 (filing date Feb. 28, 1994)

Myers et al. (Myers), "Environment for rapidly creating interactive design tools," The Visual Computer, Vol. 8, No. 2, pp. 94-116 (June 1992).

All claims on appeal, claims 1-27, stand rejected under 35 U.S.C. §103. As evidence of obviousness as to claims 1-18, the examiner relies upon appellant's admitted prior art (apparently specification page 1, line 10 through page 3, line 21),

further in view of Pazel and Berry. In the second rejection, the examiner relies upon the combination of teachings and showings in Frid-Nielsen in view of Myers to reject claims 19-27.

Rather than repeat the positions of the appellant and the examiner, reference is made to the briefs and the answer for the respective details thereof.

OPINION

Generally for the reasons set forth by the examiner in the answer, we sustain the rejection of all claims on appeal under 35 U.S.C. § 103 as embellished by the following.

As established by the examiner in the rejection, the starting point is the admitted prior art in the specification identified earlier. The basic underpinnings of object oriented programming (OOP) are established in these admitted prior art considerations of appellant. We do not, however, agree with appellant's view that Berry is not directed to OOP. Even the initial discussion in the abstract of the source object-target object interplay is suggestive of OOP. The examiner's reliance on Figure 7 (to which we would add Figure 8) to indicate that object-oriented programming is generally taught in this reference is significant since together they disclose the use of source objects, source handles, target objects, target handles, properties, etc., all of which are

indicative of terminology associated with OOP. The prior art discussion at column 1 of Berry is instructive as to setting an additional context under which Berry's teachings must be viewed. This would include user interfaces utilizing existing visual programming products to allow users to visually create simple event-action sequences with respect to source objects and target objects. These are all discussed in the context of inserting, deleting, clearing, etc., functions associated with normal editing operations by a programmer. In this sense, the context of Berry may be viewed as a program editor for modifying existing programs in OOP. Berry teaches the use of graphical user interfaces (GUIs) to link objects in OOP. Although not explicitly teaching that the GUI environment is Windows-based, Figures 2-6 are clearly suggestive of that in addition to the ability of the user to use the user interface adaptor 22 in Figure 1 by the actuation of the mouse 26 and keyboard 24 to effect the GUI functions depicted in these figures. These functions also include conventional Windows-type drag and release operations for manipulating graphical objects such as those displayed in the various figures.

On the other hand, in a more specific context, Pazel teaches that a Windows-type environment is utilized with its attendant graphical user interfaces to simplify software programming development and debugging and even to correct software

programs according to the initial paragraphs of column 1 of this patent. Various debugging software systems are taught at columns 1 and 2, even to including specific discussions of program editors which allow the user to enter or change code at least in conventional programming architectures. Although the abstract and columns 2 and 3 of Pazel discuss the use of objects, they are in the context of objects associated with GUIs and not necessarily objects in the context of object-oriented programming. In this respect, appellant's observation that Pazel is not directed to object-oriented programming is correct. Its teaching value, however, goes well beyond the specific use of Windows and GUI environments to aid the software programmer in editing and otherwise debugging programs. Obviously, in the context of the object-oriented programming disclosures of the admitted prior art and Berry relied upon by the examiner, the suggestibility of utilizing the GUIs and Windows-based approach of Pazel in the admitted prior art and Berry was compelling.

In a more powerful analytical sense, appellant's admitted prior art of OOP in combination with Berry is most compelling since they both are concerned with OOP. Pazel then may be viewed as cumulative to the teachings of GUIs utilized in a program development environment to Berry, but it has remarkably more extensive teachings of GUIs in a Windows environment for editing purposes. We recognize, in contrast to

appellant's arguments in the brief, that Pazel does not teach changing programming functions or editing them in a specific sense, but he only does so in a general sense. Therefore, appellant's focus in the arguments portion of the brief on the alleged deficiencies with respect to object-oriented programming and Pazel are misplaced. On the other hand, Berry teaches one specific editing or modifying function by the use of the linking graphical user interface. The figures illustrate the manner in which this is done.

The data file structure clause of claim 1 on appeal (and other independent claims on appeal) relate to conventional object-oriented programming data structures anyway that were well known in the art. In their own way both Berry and Pazel teach graphical user interfaces of the type broadly claimed to allow the user to select visually displayed objects and to therefore modify them either according to the specific teachings of Berry or the more general editing teachings of Pazel. The concept of operating upon ancestor and descendant objects of claim 1 on appeal, for example, has been established in the context of object-oriented programming from the admitted prior art as well as the object-oriented programming teachings of Berry. The use of the particular graphical user interface in Berry for linking displayed objects for a very specific purpose of iteration is consistent with the generalized editing/program debugging teachings of

Pazel. Moreover, Berry's showings in Figures 2-6 indicate that by the use of the object link operator, he shows that the actual underlying data structures of the data files have been changed consistent with the modifying language of representative claim 1 on appeal. Consistent with this view, for example, the Figure 3 showing and its discussion at the bottom of column 3 of Berry indicates that by clicking on the PB1 220 (push-button 1) the data from the array AR1 230 is actually inserted physically into the DB1 210 or database 210. Thus, the data file modification operation depicted in the figures is actually effected, as claimed, by the use of the GUI operators.

In the context of the generalized teachings of program editing in a Windows environment in Pazel and the use of GUIs in an object-oriented programming editor of Berry, both teachings would have been compelling to the artisan to overcome the known prior art editing and program redesign approaches of the prior art discussed at the bottom of appellant's specification page 2 and the top of page 3. There it is clear that it is known in the art to perform such modifying or editing operations within the context of object-oriented programming but in a very cumbersome manner. The graphical user interface approaches of Berry and Pazel clearly would have been an obvious ideal methodology by which the user obviously would have overcome these known prior art difficulties. The broadly defined modifying of independent claim 1 and

the more specific concept of addition and splicing of claims 3 and 4 are known editing operations in a general program editing environment such as in Pazel as well as the admitted prior art with respect to OOP. The ability to change the selected properties and handlers of claim 5 also is a known concept in editing prior art object-oriented programs.

With respect to these overall considerations as a whole of the admitted prior art, Berry and Pazel, the features recited in dependent claim 8 and independent claim 9 relating to a particular window approach with which to perform the editing operations of the claimed invention clearly would have been obvious to the artisan as the most ideal setting in which to perform the editing operations. In this respect, because Pazel deals specifically in a Windows environment, his teachings and showings are most telling. The specific recitation of a first, second and third panel relating to a corresponding object of interest, its parent and any children objects thereof, come initially from the normal editing operations of the admitted prior art as to OOP. Even so, in a more traditional programming context, even Figure 2 of Pazel depicts the debug program within the RAM 16 in Figure 1 in a Windows environment having different portions each of which may be selectively considered a panel which may be operated upon by the user. For example, the source file 2 in Figure 3 and its particular Windows based-

suboperations for Run and Options of corresponding Figures 4 and 5 are put together in Figure 7 utilizing additionally the source file N as a separate window to be added thereto. The examiner's reliance upon a CALL function in this reference is probably over emphasized by the examiner, while still illustrating that the called function in Figure 6 may be associated in a single window in Figure 7 with its source file 2. Based on the teachings of Pazel, it clearly would have been obvious to the artisan to have utilized the well knownl data structures of the admitted prior art relating to object oriented programming to depict panels or separate windows as well of particular objects of interest, their respective parent and children objects of the type broadly set forth in claims 8 and 9 on appeal. We are therefore unpersuaded of appellant's arguments in the brief and reply brief as to these claims. Finally, the features of independent claim 15 have already been addressed with respect to our consideration of earlier claims.

We turn to the rejection of claims 19-27 under 35 U.S.C. § 103 relying upon the combined teachings and showings in Frid-Nielsen in view of Myers. Our study of these references is consistent with the examiner's views expressed in the statement of the rejection portion of the answer as well as the responsive arguments portion of it. This latter section of the answer appears to us to directly address the arguments raised by appellant at page 11 of the brief. The examiner admits, and we agree, that Frid-Nielsen

does not teach that his integrated development environment alters or edits but merely permits the inspection or viewing or browsing of object-oriented programming environments as set forth initially in the title of the invention and in the abstract. Significantly, however, columns 1-3 of Frid-Nielsen detail very persuasively in an historical sense the development and desirability of using GUIs in a Windows environment as a tool in the development of object-oriented programming methodologies. Note particularly the paragraph bridging columns 2 and 3. The examiner's reliance upon Figure 7G of this reference is significant and consistent with his reliance thereon. We would add to this Figures 7H, 7I, the showing in Figures 7N of parent and child relationships, the showing in Figure 8A of parent and child relationships, the discussion and showing of handles in Figure 8B, the showing of parent and child relationships in Figure 8C and the Figure 8D showing of the mouse-keyboard user actuatable device for manipulating the inspection browsing software of Frid-Nielsen.

We do not agree with appellant's characterization of Myers at page 11 of the brief that it does not teach the use of a graphical user interface for modifying program structures with respect to hierarchical or heteroarchical relationship of objects. What appellant makes reference to at page 102 is the discussion of what Myers considers to

be prior art to him. In fact, the entire discussion of topic 5.1 beginning at column 1 of page 101 through the discussion of topic 5.1.3.2 at the second column of page 103 is most telling, as the examiner emphasizes in the responsive arguments portion of the answer. There is even an emphasis in this discussion of structural inheritance such as that depicted in Figures 6a, b. The discussion between the noted pages is concerned with inserting, deleting, changing inheritances and relationships between parent and child objects in an object-oriented programming environment utilizing Windows and graphical user interfaces as an interaction design tool upon object-oriented programming data structures. We are thus unpersuaded of appellant' arguments with respect to independent claim 19 and dependent claim 20. Moreover, no arguments at all have been presented as to independent claim 24 and its dependent claims in the brief and reply brief.

In view of the foregoing, we affirm the examiner's rejections of claims 1-18 and 19-27 as separately stated rejections under 35 U.S.C. §103. Therefore, the decision of the examiner rejecting claims 1-27 under 35 U.S.C. §103 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 CFR § 1.136(a).

AFFIRMED

James D. Thomas Administrative Patent Judge	))
	) BOARD OF PATENT
Errol A. Krass Administrative Patent Judge	)) APPEALS AND
	) INTERFERENCES
Michael R. Fleming Administrative Patent Judge	))

JDT/cam

Appeal No. 1999-1452
Application 08/436,830

James W. Peterson
BURNS, DOANE, SWECKER & MATHIS
P. O. Box 1404
Alexandria, VA 22313-1404